

Mata Kuliah : Pemrograman Berorientasi Objek
Dosen Pengampu : Novan Zulkarnain, S. T., M. Kom.
Nama : Anggia Susanti
NIM : 2222105233
Kelas : 2TI03

UJIAN AKHIR SEMESTER (UAS) MEMBUAT GAME DARI JAVA GAME SNAKE MENGGUNAKAN JAVA (VSCODE)

Game snake yang saya buat ini adalah game klasik yang dimana pemain mengendalikan ular yang terus bergerak di dalam sebuah area menggunakan panah (*right, left, down dan up*) pada *keyboard*, dengan memakan-makanan yang muncul secara random dan akan bertambah panjang ularnya setiap memakan makanan tersebut. Selain bertambah panjang, kamu juga akan mendapatkan skor 1 setiap memakan makanan tersebut.

Tujuan dari game ini adalah bertahan selama mungkin tanpa menabrak dinding area permainan atau badan ular sendiri. Jika ular tersebut menabrak dinding area permainan atau badannya sendiri, maka permainan berakhir (*game over*).

Cara membuat permainan ular di Java menggunakan pustaka grafis awt dan swing dan menggunakan antarmuka pengguna grafis (GUI) untuk menampilkan game Snake, membuat game loop, mendengarkan penekanan tombol untuk menggerakkan ular, menggunakan ArrayList untuk menyimpan segmen ular, dan memeriksa tabrakan dan di luar batas untuk permainan berakhir

Didalam folder 'src' memiliki file 'app.java' yang didalam nya membuat jendela GUI dengan ukuran 600x600 piksel yang menampilkan SnakeGame. Jendela ini tidak bisa diubah ukurannya dan SnakeGame diatur untuk menerima input dari pengguna setelah jendela ditampilkan.

```
import javax.swing.*; //mengimpor seluruh Pustaka Swing dari java
yang digunakan untuk membuat antarmuka pengguna grafis (GUI)

//class utama dari program app.java
public class App {
    public static void main(String[] args) throws Exception {
```

```

        //memiliki papan 600x600 piksel
        int boardWidht = 600;        //lebar papan permainan 600 piksel
        int boardHeight = boardWidht; //tinggi papan permainan =
lebar papan permainan = 600 piksel

        //JFrame digunakan untuk membuat jendela dengan judul yang
diinginkan
        JFrame frame = new JFrame("SNAKE GAME_Anggia
Susanti_2222105233");
        frame.setVisible(true); //supaya jendela terlihat
        frame.setSize(boardWidht, boardHeight); //mengatur ukuran
jendela sesuai dengan variable 'boardWidht' dan boardHeight'
        frame.setLocationRelativeTo(null); //penempatan jendela di
tengah layar
        frame.setResizable(false); //mencegah ukuran jendela diubah
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //operasi untuk menutup jendela ketika mengklik tombol x yaitu
keluar dari aplikasi

        SnakeGame snakeGame = new SnakeGame(boardWidht, boardHeight);
        //membuat instance dari kelas 'SnakeGame' dengan ukuran papan yang telah
ditentukan
        frame.add(snakeGame); //menambahkan instance 'SnakeGame' ke
dalam 'JFrame'
        frame.pack(); //menyesuaikan ukuran JFrame
        snakeGame.requestFocus(); //focus pada komponen
'snakeGame' untuk menerima input dari keyboard
    }
}

```

Setelah itu buat new file didalam folder 'src' dengan nama file 'Game.java' (opsional) untuk file permainan SnakeGame tersebut.

```

import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList; //untuk menyimpan segmen tubuh ular
import java.util.Random;    //untuk mendapatkan nilai X dan Y secara
random untuk menempatkan makanan ular dilayar
import javax.swing.*;

//class 'SnakeGame' yang mengatur permainan. Class ini dibuat untuk
menangani aksi dan input keyboard
public class SnakeGame extends JPanel implements
ActionListener, KeyListener{

```

```

    //class Tile mempresentasikan posisi x dan y dipapan permainan
private class Tile {
    int x;
    int y;

    Tile(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

    //menyimpan lebar dan tinggi papan permainan
int boardWidht;
int boardHeight;
int tileSize = 25;    //menyimpan ukuran tiap kotak dipapan
permainan

    //Snake
    Tile snakeHead;    //menyimpan posisi kepala ular
    ArrayList<Tile> snakeBody;    //menyimpan bagian-bagian tubuh
ular ketika bertambah saat memakan makanannya

    //food
    Tile food;    //menyimpan posisi makanan
    Random random;    //menempatkan posisi makanan secara random

    //game logic
    Timer gameLoop;    //timer yang mengatur interval untuk memanggil
aksi permainan
    int velocityX;    //kecepatan pergerakan ular
    int velocityY;    //kecepatan pergerakan ular
    boolean gameOver = false;    //menyimpan status game apakah
sudah berakhir atau belum

//mengatur ukuran, background dan property lainnya
//menginisialisasi kepala ular, tubuh ular, makanan dan logika
permainan
    SnakeGame(int boardWidht, int boardHeight) {
        this.boardWidht = boardWidht;
        this.boardHeight = boardHeight;
        setPreferredSize(new Dimension(this.boardWidht, this.
boardHeight));
        setBackground(Color.black);    //menentukan warna background
        addKeyListener(this);    //mengunci tombol penggerak ular
        setFocusable(true);    //memfokuskan tombol penggerak
untuk ular

```

```

        snakeHead = new Tile(5, 5);
        snakeBody = new ArrayList<Tile>();

        food = new Tile(10, 10);
        random = new Random();
        placeFood();

        velocityX = 0;
        velocityY = 0;

        //memulai dengan interval 100 ms
        gameLoop = new Timer(100, this);
        gameLoop.start();
    }

    //untuk menggambar komponen pada permainan
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        draw(g);
    }

    public void draw(Graphics g) {

        //menentukan warna makanan
        //food
        g.setColor(Color.red);
        g.fill3DRect(food.x * tileSize, food.y * tileSize, tileSize,
tileSize, true);

        //menentukan warna kepala ular
        //snake head
        g.setColor(Color.blue);
        g.fill3DRect(snakeHead.x * tileSize, snakeHead.y * tileSize,
tileSize, tileSize, true);

        //untuk menambahkan tubuh ular setiap memakan makanannya
        //snake body
        for (int i = 0; i < snakeBody.size(); i++) {
            Tile snakePart = snakeBody.get(i);
            g.fill3DRect(snakePart.x * tileSize, snakePart.y * tileSize,
tileSize, tileSize, true);
        }

        //score
        g.setFont(new Font("Arial", Font.PLAIN, 16));
        if (gameOver) {
            g.setColor(Color.red);    //menentukan warna pesan game over
            dan score yang didapatkan

```

```

        g.drawString("Game Over: " +
String.valueOf(snakeBody.size()), tileSize - 16, tileSize); //pesan
"Game Over : (hasil score yang didapatkan)"
    }
    else {
        g.drawString("Score: " + String.valueOf(snakeBody.size()),
tileSize - 16, tileSize); //tampilan "Score : (hasil score, dan
akan bertambah ketika berhasil memakan makanannya tanpa game over)"
    }
}

//mengacak posisi makanan dipapan permainan dan mengubah posisi
makanan secara random ketika ular berhasil memakan makanan diposisi
yang lain
public void placeFood() {
    food.x = random.nextInt(boardWidht/tileSize); //600/25 = 24
    food.y = random.nextInt(boardHeight/tileSize);
}

//memeriksa apakah dua tile berada pada posisi yang sama
public boolean collision(Tile tile1, Tile tile2) {
    return tile1.x == tile2.x && tile1.y == tile2.y;
}

public void move() {
    //untuk memakan makanan jika ular bertabrakan dengan makanan
maka makanan hilang dan berpindah tempat. Kemudian, tubuh ular akan
bertambah
    //eat food
    if (collision(snakeHead, food)) {
        snakeBody.add(new Tile(food.x, food.y));
        placeFood();
    }

    //Snake body
    //menggambar tubuh ular yang bertambah dan mengulangi kondisi
tersebut dan tubuh tersebut mengikuti kepala ular
    for (int i = snakeBody.size()-1; i >= 0; i--) {
        Tile snakePart = snakeBody.get(i);
        if (i == 0) {
            snakePart.x = snakeHead.x;
            snakePart.y = snakeHead.y;
        }
        else {
            Tile prevSnakePart = snakeBody.get(i-1);
            snakePart.x = prevSnakePart.x;
            snakePart.y = prevSnakePart.y;
        }
    }
}

```

```

    }

    //kecepatan x dan y
    //Snake Head
    snakeHead.x += velocityX;
    snakeHead.y += velocityY;

    //game over conditions
    //game over jika kepala ular bertabrakan dengan tubuhnya
    sendiri
    for (int i = 0; i < snakeBody.size(); i++) {
        Tile snakePart = snakeBody.get(i);
        //collide with the snake head
        if (collision(snakeHead, snakePart)) {
            gameOver = true;
        }
    }

    //game over jika kepala ular menabrak dinding pada papan
    permainan
    if (snakeHead.x*tileSize < 0 || snakeHead.x*tileSize >
boardWidht ||
        snakeHead.y*tileSize < 0 || snakeHead.y*tileSize >
boardHeight) {
        gameOver = true;
    }
}

@Override
public void actionPerformed(ActionEvent e) {
    move();
    repaint();
    //jika game over permainan akan berhenti
    if (gameOver) {
        gameLoop.stop();
    }
}

//menentukan 4 tombol untuk pergerakan ularnya (UP atas, DOWN
bawah, RIGHT kanan dan LEFT kiri)
@Override
public void keyPressed(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_UP && velocityY != 1) {
        velocityX = 0;
        velocityY = -1;
    }
    else if (e. getKeyCode() == KeyEvent.VK_DOWN && velocityY != -1)
{

```

```

        velocityX = 0;
        velocityY = 1;
    }
    else if (e.getKeyCode() == KeyEvent.VK_LEFT && velocityX != 1) {
        velocityX = -1;
        velocityY = 0;
    }
    else if (e.getKeyCode() == KeyEvent.VK_RIGHT && velocityX != -1)
{
        velocityX = 1;
        velocityY = 0;
    }
}
}
}

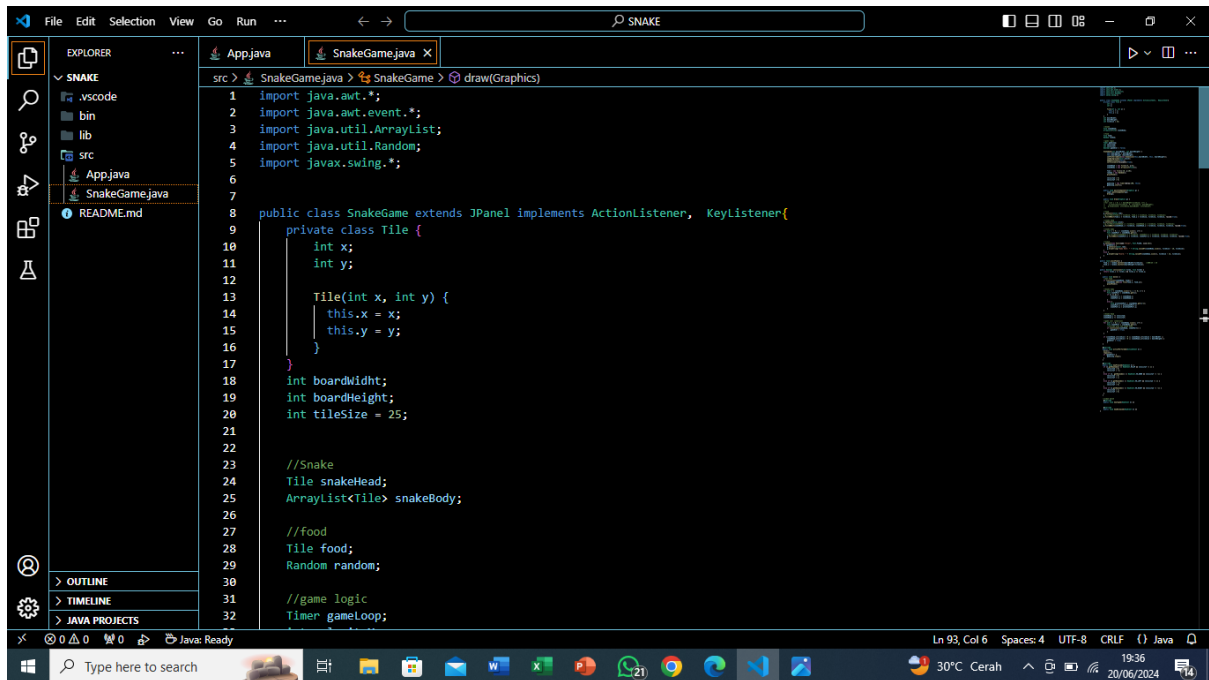
```

Keterangan:

- Penjelasan code dijelaskan didalam code diatas dengan komentar (//) berwarna abu-abu
- Sumber pembuatan snake game: <https://youtu.be/Y62MJny9LHg?si=g-ZGvuCw4O14r-AB>

LAMPIRAN

Src SnakeGame:



```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.util.ArrayList;
4 import java.util.Random;
5 import javax.swing.*;
6
7
8 public class SnakeGame extends JPanel implements ActionListener, KeyListener{
9
10     private class Tile {
11         int x;
12         int y;
13
14         Tile(int x, int y) {
15             this.x = x;
16             this.y = y;
17         }
18     }
19
20     int boardWidth;
21     int boardHeight;
22     int tileSize = 25;
23
24     //Snake
25     Tile snakeHead;
26     ArrayList<Tile> snakeBody;
27
28     //food
29     Tile food;
30     Random random;
31
32     //game logic
33     Timer gameLoop;
```

Tampilan ketika baru memulai game:



Tampilan ketika game selesai atau game over:

